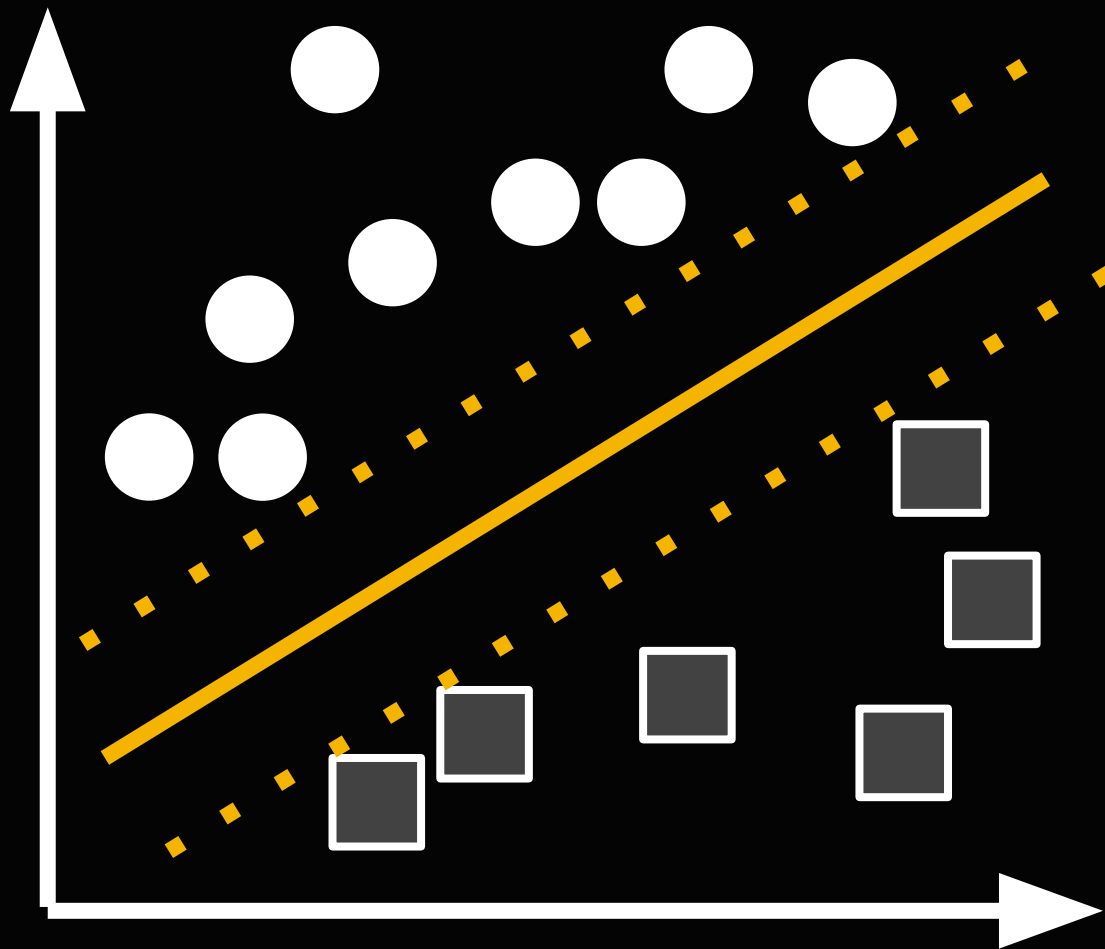
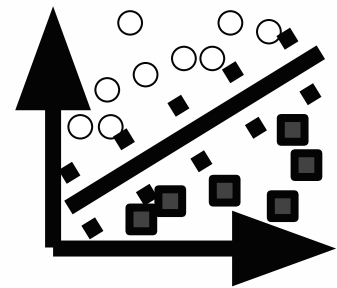
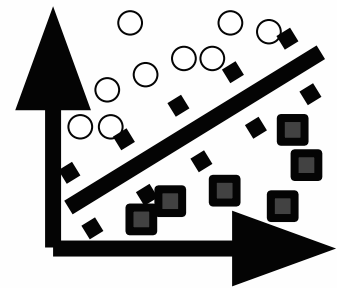


What are
SVM
kernels



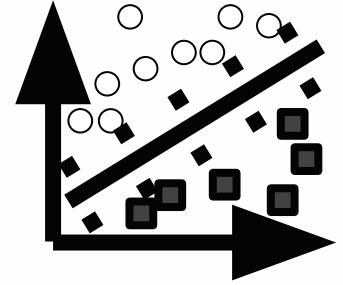


With SVM, it's **easy to make** a linear
hyperplane between classes



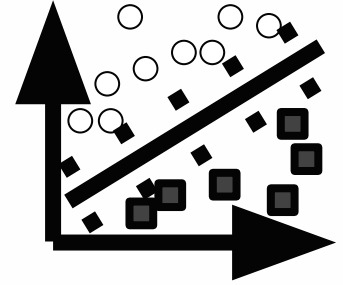
- Kernel strategy is used often to turn fast linear methods into **fast nonlinear methods**

SVM kernel

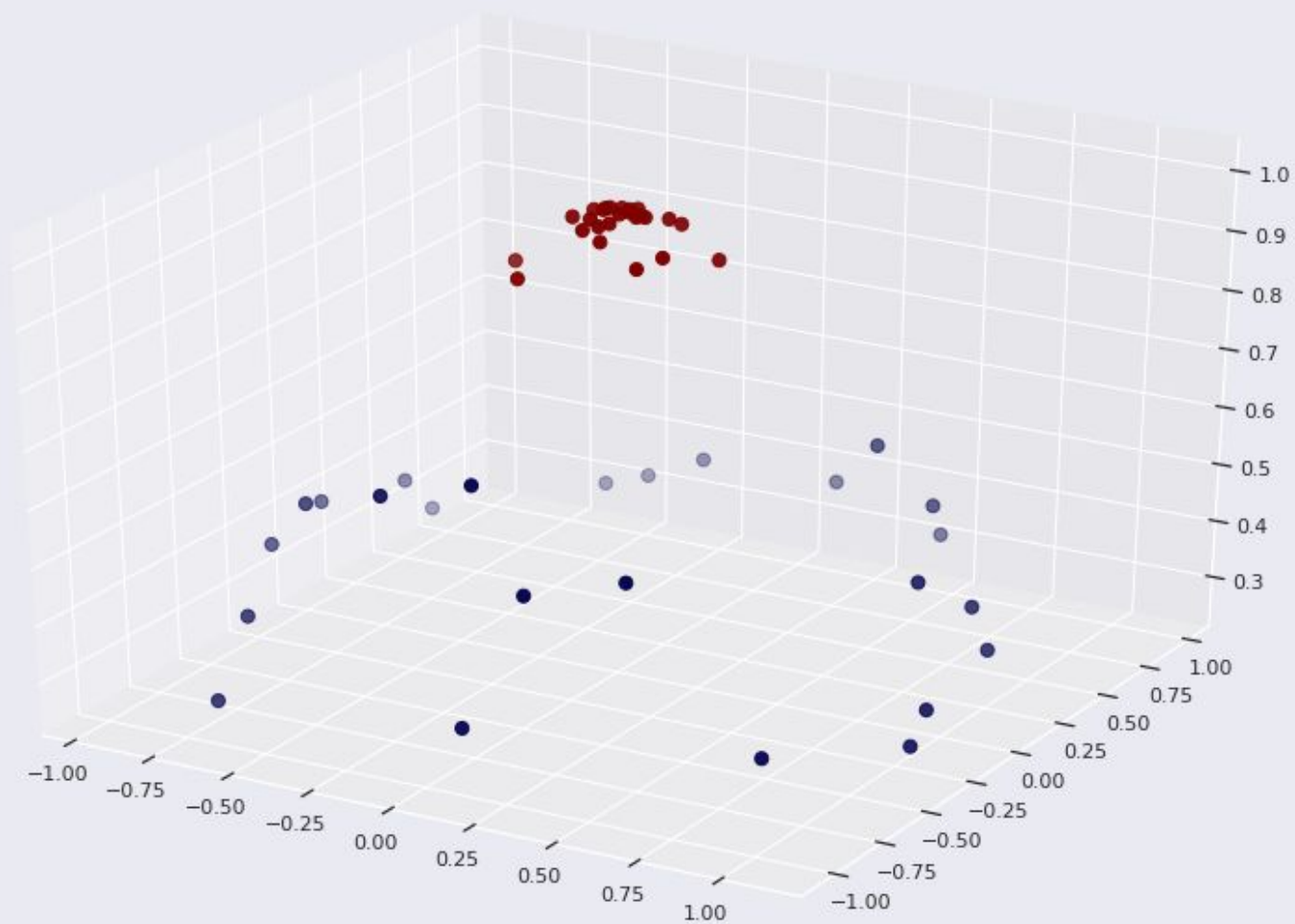


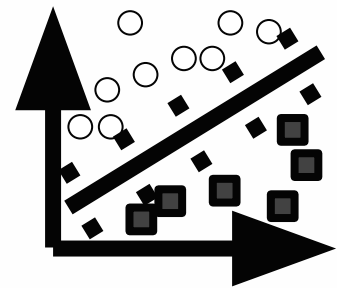
- **finds the method to separate data points** based on the target classes

SVM kernel

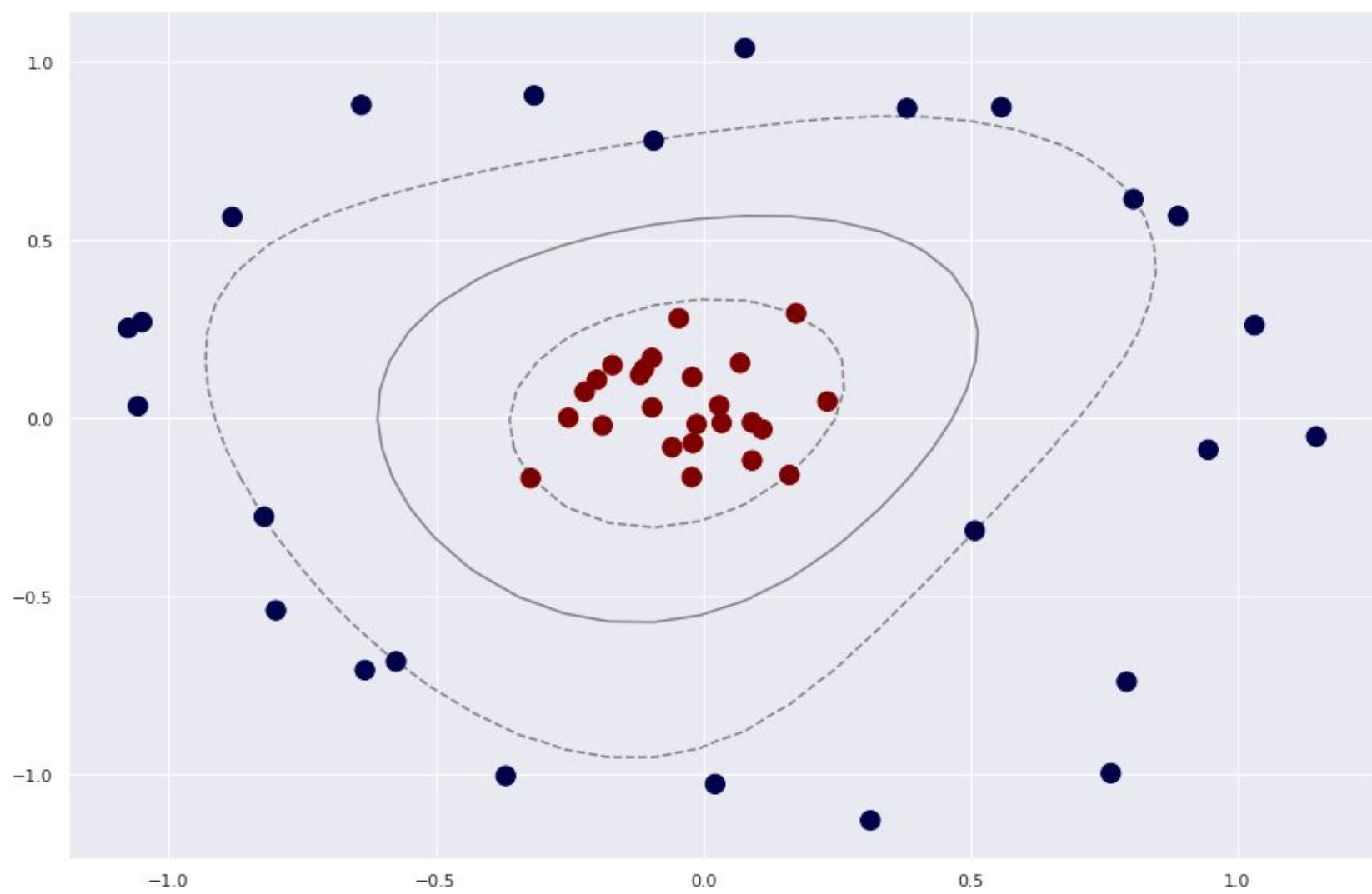


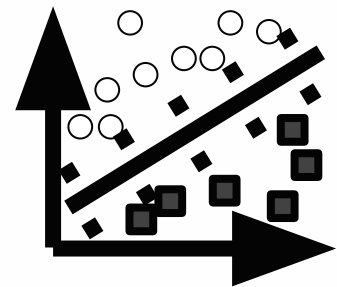
- function that **transforms low dimensional input** into a better dimensional space
 - converts non-separable problems to separable problems.





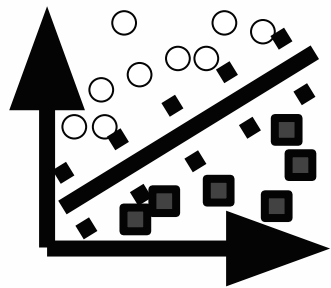
- With an **extra dimension**, the data is **easy to separate** linearly
 - Draw a **separating plane**



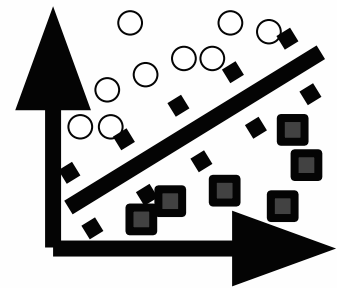


- We had to **choose and tune** our kernel.
- If you don't center our radial basis function in the right location, you won't get such **clean results**.

Problem

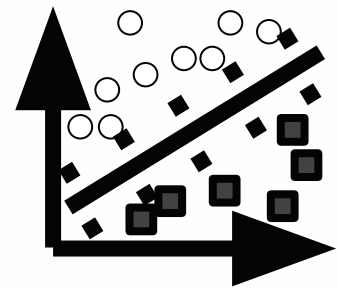


- How do we know **which equation** to choose?
- We need to **quickly find** the best kernel.

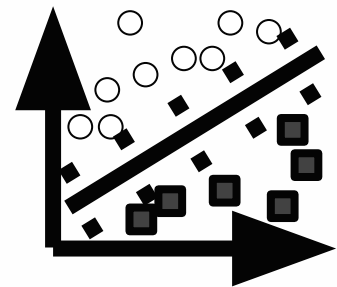


Solution

- **Try a kernel** centered at *every point* in the dataset
- Use the kernel with the **best results**

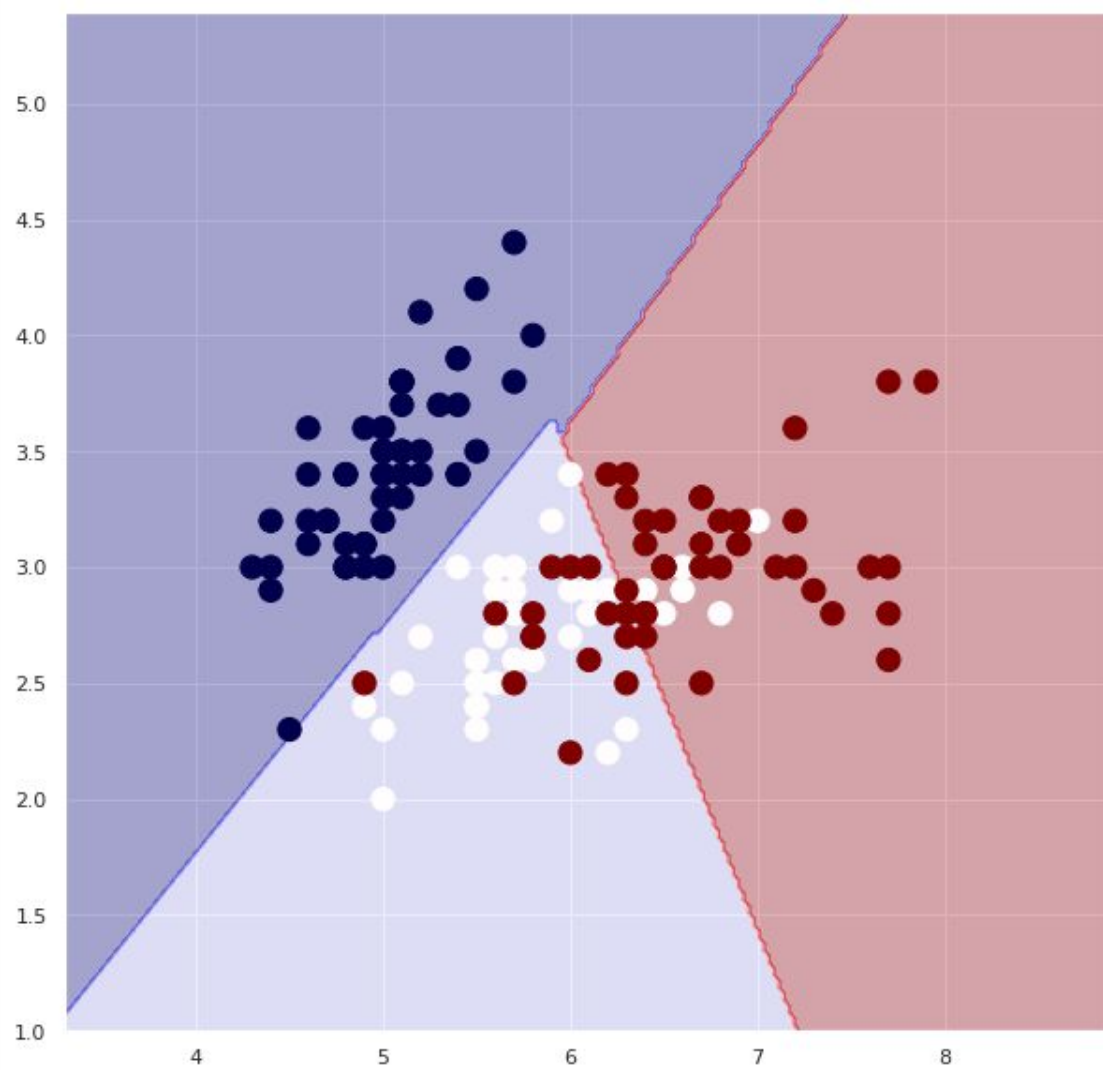


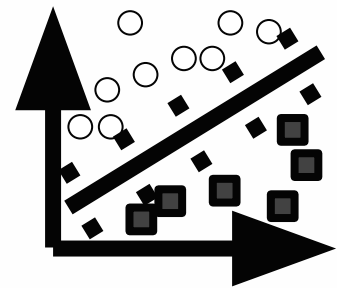
Trying different kernels can be
computationally intensive



With the “Kernel Trick”, **kernel transformation** can be done implicitly

- **Built into SVM!**
- Change the **kernel type** from linear to RBF (Radial Basis Function)





Linear Kernel

- most **basic** type of kernel
- usually **1D** in nature
- Best for a dataset with **a lot of features**
- Best for classes that can be **linearly separated**
- Faster than others

*Find more kernels from
the Scikit library*



What are
SVM
kernels ✓

